(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 201 330**
A2

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 86303486.4

(22) Date of filing: 08.05.86

(51) Int. Cl.⁴: G 11 B 20/18

(30) Priority: 08.05.85 US 732353

(43) Date of publication of application: 12.11.86
Bulletin 86/46

(84) Designated Contracting States: AT BE CH DE FR GB IT LI LU NL SE

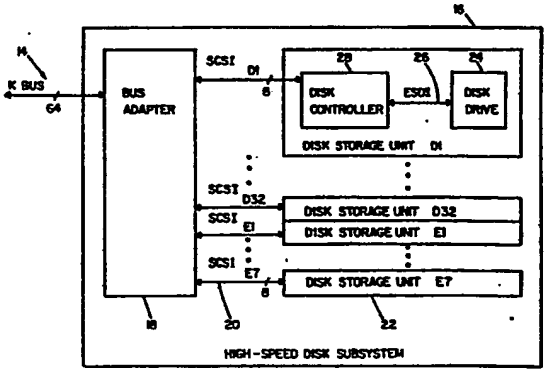
(71) Applicant: THINKING MACHINES CORPORATION, 245 First Street, Cambridge Massachusetts 02142 (US)

(72) Inventor: Hillis, Daniel W., 548 Greene Street, Cambridge Massachusetts (US)
Inventor: Liu, Clement K., 41 Claymoss Road, Brighton Massachusetts (US)

(74) Representative: Deans, Michael John Percy et al, Lloyd Wise, Tregear & CO. Norman House 105-109 Strand, London WC2R OAE (GB)

(54) Apparatus for storing digital data words.

(57) A storage system for data words generates error correction bits for each data word which are stored independently from the data word on a separate mechanically-driven medium, in one aspect of the system. In another aspect, the storage system serves a wide high throughput parallel bus by storing different portions of each data word that appears on the bus in different asynchronous storage units.



EP 0 201 330 A2

0201330

APPARATUS FOR STORING DIGITAL DATA WORDS

This invention relates to storing digital data.

In a typical magnetic disk drive, several hard magnetic disks are mounted on a single rotating spindle. The storage space on each disk surface is organized in concentric tracks. The corresponding tracks on all of the disk surfaces form an imaginary cylinder. Each disk surface is served by a read/write head. All of the read/write heads can be moved together to any selected cylinder and can then store or retrieve data on the tracks that make up that cylinder.

Normally the magnetic disk drive is connected by a serial bus to a disk controller that, among other things, directs the read/write heads to move to the proper track, buffers data that is about to be stored on or has just been retrieved from the disk, ensures that the data is stored or retrieved at the proper address, and performs serial error correction and detection on the data. Typically the disk controller serves a parallel input/output bus connected to a computer whose input/output port is, for example, 32 bits wide. Thirty-two bit words delivered over the bus are passed via the disk controller and serially loaded onto or retrieved from the disk drive. The throughput rate of the serial bus between the disk controller and the disk drive is roughly matched to the throughput rate of the parallel I/O bus between the computer and the disk controller.

In order to expand the total available storage space, multiple disk driver can be linked to a single controller in a daisy chain or star configuration.

One general feature of systems described herein is in generating error correction bits for the data words and storing the correction bits independently from the data words on a separate mechanically-driven medium.

The present invention provides, in a first aspect thereof, apparatus for storing and retrieving digital data words and including data storage equipment adapted for storing each said data word on a mechanically-driven medium, said apparatus being characterised in that it comprises error correction means adapted for correcting errors in retrieved data words, which error correction means comprise: error correction circuitry adapted to generate error correction bits for said data words, and correction bit storage equipment adapted for storing said correction bits independently from said data word on a mechanically driven medium, said error correction circuitry being arranged operatively to correct errors in retrieved said data words using said error correction bits.

Preferred embodiments of the system include the following features: Different portions of each data word are stored in different data storage units and there is at least one correction bit storage unit for storing the correction bits. Errors are corrected in a manner that is tolerant to the failure of any single storage unit. Various ones of the error correction bits for a given data word are stored in different correction bit storage units (preferably with no two error correction bits for a given data word being stored in a single storage unit). A set of error correction bits are generated for a group of fewer than all of the bits in a data word at one time, and groups of the data word bits are processed in succession to produce successive sets of error correction bits. Alternatively, the different groups of data word bits are processed in parallel at the same time to generate the sets of error correction bits. For example, each data word could have 64 bits making up two groups of 32 bits each, and a set of 7 correction bits could be generated for each group of 32 bits. The data words are delivered in succession via a parallel data bus for storage and retrieval, and there is an adaptor connected between the bus and the storage units to route the data words between the bus and the storage

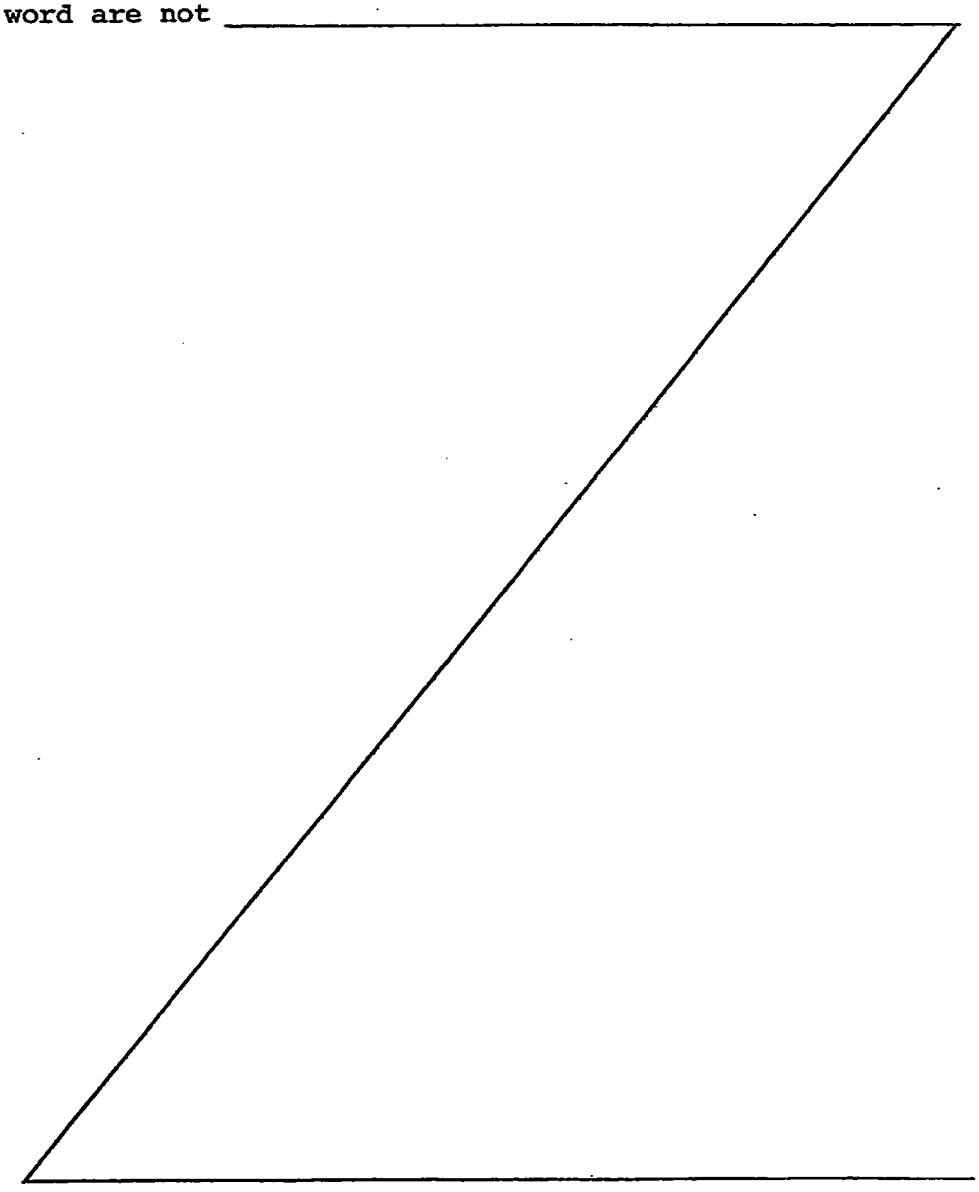
units.

Another general feature of systems described herein is to store different portions of each data word that appears on the parallel bus in different asynchronous mechanically-driven storage units.

According to a second and alternative aspect of the present invention, there is provided apparatus for storing digital data words and adapted for delivering them in succession to a parallel data bus when requested by a data processor connected to said bus, said apparatus being characterised in comprising: storage units each having a mechanically-driven medium, and a head for reading said medium, the storage units being so arranged that different portions of each said data word are stored in different said storage units, one said storage unit having associated therewith a first delay time between when a particular said data word is requested and when the portion of said particular data word stored in said storage unit is ready to be delivered to said bus, another said storage unit having associated therewith a second delay time, different from said first delay time, between when said particular data word is requested and when the portion of said data word stored in said other storage unit is ready to be delivered to said bus; and an adaptor arranged for operative connection between said bus and said storage units and adapted to respond to said request from said data processor by causing said storage units to read said portions of each successive said word from said storage units.

Preferred embodiments of the system include the following features: A data word is retrieved by delaying the delivery of any portion of the data word to the bus until all portions of the data word are ready to be delivered. Each storage unit sends a drain command when it is ready to deliver its portion of the requested data word, and the storage units are drained only when all

storage units have issued drain commands. Parallel error correction bits are generated for each data word and at least one of the error correction bits is stored on a storage unit that does not hold any portion of the related data word. The error correction bits are read at times corresponding to the reading of the data word and are used for correcting single bit errors in the data word. When storing data words, the different portions of the data word are not



delivered to the different storage units until they are all ready to store. Each storage unit has a buffer and successive data words to be read out are requested at a rate sufficient to assure that no buffer becomes full; and successive words to be written are delivered at a rate sufficient to assure that no buffer becomes empty. Equal numbers of bits (e.g., 2 bits) of each data word are stored in different storage units. Each individual bus serving a storage unit conforms to the SCSI protocol. There are more than 32 bits (e.g., 256 bits) in each data word. The data words are delivered to the parallel bus at a rate of at least 4 megabits per second per line of the bus. Each storage unit is a magnetic disk drive.

We can provide a very high throughput rate storage system capable of serving a very wide, high-speed parallel I/O bus. Standard commercially available disk drives and controllers can be used without having to synchronize the rotations of the different disk drive spindles. By storing error correction bits and data bits independently, the system is tolerant to the failure of any one entire disk drive unit. Standard SCSI protocol commands are used to synchronize the overall operation of the system.

Other advantages and features of the systems will become apparent from the following description of preferred embodiments.

We first briefly describe the drawings.

Fig. 1 is a block diagram of a computer and related storage system.

Fig. 2 is a block diagram of one high-speed disk subsystem of Fig. 1.

Fig. 3 is a block diagram of a representative portion of the bus adaptor circuitry of Fig. 2.

Fig. 4A, 4B are timing diagrams of data store and data retrieve operations.

Fig. 5 is alternate circuitry to Fig. 3.

Referring to Fig. 1, a computer 10 (of the kind described in our European Patent Application No. 84303598.1, the disclosure of which is to be regarded as incorporated herein by reference) has four identical input/output (I/O) ports 12. Each I/O port 12 is connected via a 64-bit wide parallel bus 13 (KBUS) to a corresponding high-speed disk subsystem 16 for storing data received over the KBUS, and for delivering retrieved data to the KBUS. Each KBUS has a throughput rate of 32 megabytes per second (8 bits per byte); data can be passed over all four KBUSes synchronously at an aggregate throughput rate of 128 megabytes per second.

Referring to Fig. 2, each disk subsystem 16 includes a bus adapter 18 connected to one of the KBUSes. The bus adaptor 18 is also connected via 39 different parallel buses 20 (SCSI buses), each 8-bits wide, to 39 different disk storage units 22. Each disk storage unit 22 includes a disk drive 24 (e.g., Control Data Model 9415) connected via a serial bus 26 (ESDI bus) to a disk controller 28 (e.g., Champion model sold by Emulex) which is in turn connected to the SCSI bus that serves that disk storage unit. Each SCSI bus carries data and commands (in the form of command descriptor blocks--CDBs) in accordance with the Small Computer System Interface standard defined in U.S. Department of Commerce, National Bureau of Standards Publication X3T9 2/82-2, Revision 14B, November 6, 1984, the disclosure of which is to be regarded as incorporated by reference.

The 39 disk storage units and the 39 SCSI buses are organized into two groups. Thirty-two of the disk storage units and buses (denoted D1 through D32) are used to store and deliver data from computer 10. The remaining 7 disk storage units and buses (denoted E1 through E7) are used to store and deliver error correction bits that are derived from the data bits in a manner to be described below, and are used for error correcting. Except for this distinction in the type of information they carry, all 39 disk storage units and SCSI buses are identical.

Each ESDI bus carries data and commands serially in accordance with the Enhanced Small Device Interface defined for example in Magnetic Peripherals, Inc. publication 77738076-D, 1984, the disclosure of which is to be regarded as incorporated by reference.

Referring to Fig. 3, in each bus adaptor 18, the 64 lines of the KBus are split into two sets of 32 lines each, the two sets being connected respectively to the bidirectional WD/CD inputs (write data or corrected data) of two error detection and correction units 32 (each unit comprising, for example, a pair of Intel 8206 or AMD 2960 chips). The 32 lines connected to each WC/CD input are also connected respectively to the inputs of 32 eight-bit shift registers 34. Each shift register thus has its two-bit input connected to one line from each of the two sets of 32 lines, and each shift register is arranged to shift two bits at a time.

For each set of 32 data bits applied to the WD/CD input of each unit 32, the unit generates seven parallel error correction bits that provide information sufficient to correct any single bit error in any of the related 32 data bits that occur during subsequent retrieval of those bits.

The seven error correction bits of each unit 32 are delivered from the SC output via seven error correction bit lines that are connected respectively to the inputs of seven eight-bit shift registers 36 that are like registers 34.

After one transfer cycle of the KBUS has elapsed, each shift register 34 contains 2 data bits corresponding to the data bits appearing on two of the lines of the KBUS, and each shift register 36 contains two of the corresponding error correction bits. After four transfer cycles of the KBUS have elapsed, each shift register 34 contains 8 data bits (one data byte) and each shift register 36 contains 8 error correction bits (one error correction byte). At that time the bytes in the shift registers 34, 36 are unloaded respectively onto SCSI buses D_1-D_{32} and E_1-E_7. Then the process is repeated for the next four transfer cycles of the KBUS, and so on. The loading and unloading of the shift registers 34, 36 are timed so that data that appears at 32 megabytes per second on the KBUS (4 megabits per line) is delivered to 32 of the 8-bit wide SCSI buses at a rate of 1.5 megabytes per second per SCSI bus. Conversely data that appears at 1.5 megabytes per second per SCSI bus is delivered to the KBUS at 32 megabytes per second.

When data is retrieved from the disk drives 24, in each byte transfer cycle of the SCSI buses, one eight-bit byte is loaded in parallel into each shift register 34, 36. Next, all of the shift registers 34 together unload the first and second bits of their error correction bytes in parallel to the RD (read data) inputs of EDC units 32, and all of the shift registers 36 unload the first and second bits of their data bytes in parallel to the CB (check bit) inputs of EDC units

32. EDC units 32 uses the error correction bits received via their CB inputs to check and correct any single bit error, appearing in the data bits received at their RD inputs. EDC units 32 then deliver the 64 corrected data bits at their WD/CD outputs to the KBUS. The process is repeated until all eight bits of the bytes held in shift registers 34, 36 are unloaded. Then the shift registers are reloaded with new bytes from the SCSI buses.

Each disk controller 28 (Fig. 2) is arranged to control the storage and retrieval of data on disk drive 24 as follows.

To initiate a storage operation, computer 28 sends an appropriate command descriptor block (CDB) over the KBUS indicating that a specified number of bytes (word count) are to be stored beginning at a specified disk address. The CDB is passed by bus adaptor 18 via the SCSI bus to disk controller 28. The controller will then, if necessary, send a seek command over the ESDI bus to the disk drive to cause it to move the read/write heads to the cylinder containing the storage location whose address was specified in the CDB. The controller 28 will next request that the data to be stored be delivered over the SCSI bus. The request is passed back to the computer via the KBUS. The data is then passed over the KBUS via the bus adaptor to the SCSI bus (as previously described) and as the data bits are received they are stored in a 14

kilobyte buffer in the controller. Since each track has a capacity of about 10 kilobytes, the buffer is able to hold more than a full track worth of data at one time. The disk controller 28 will begin to unload its buffer serially onto the ESDI bus either when the buffer is full (if the word count is greater than or equal to the buffer capacity), or otherwise as soon as the number of bytes in the buffer equals the word count. The data is then stored beginning at the specified address on the track. If the word count is greater than the capacity of the buffer, then as soon as the buffer has been drained to a level where only 8 kbytes of data remain in the buffer, the controller sends a request for additional data back to computer 10 and the additional data is loaded into the buffer at the same time that the oldest data in the controller is being unloaded to disk drive 24. The process continues until the word count has been reached.

Similarly, to initiate a retrieval operation, computer 28 sends a CDB indicating that a specified number of bytes are to be retrieved beginning at a specified disk address. The controller will first, if necessary, cause the disk drive read/write heads to seek to the proper track, and to read data beginning at the designated address. The data passes serially over the ESDI bus and is loaded into the controller's serial buffer. As soon as the buffer contains a number of bytes equal to the number of bytes in a sector of the track, the controller

will send a request over the SCSI bus to computer 10 to take the data from the buffer. While that data is being unloaded, the controller continues to accept data over the ESDI bus from the disk drive and to load it into its buffer. The process continues until the number of bytes equals the word count.

The storage and retrieval of data via all of the SCSI buses to the controllers occurs synchronously in parallel. That is, from the point of view of the bus adaptor, related data is passed in a single operation over all of the SCSI buses at the same time. Since the different disk drives 24 are typically rotating out of phase with one another, it is unlikely that the corresponding track locations on different disk drives will appear under their respective read/write heads at the same time. Accordingly there will be differences in the times when different controllers are able to execute storage commands, and differences in the times when different controllers will be ready to deliver retrieved data. The potentially out-of-phase operation of the different controllers is reconciled with the need for synchronous transfers to and from the bus adaptor in the following way.

Referring to Fig. 4A, when data is to be stored, computer 10 first sends to the bus adaptor a seek command (60) naming a particular cylinder. The bus adaptor then broadcasts the seek command by passing appropriate CDBs via the SCSI buses to all of the controllers (62) to cause them all to seek to the

same cylinders. While the controllers are independently executing the seek commands, the computer can perform other unrelated operations.

After passage of a period of time (64) which is sufficient to ensure that all controllers have executed the seek command, computer 10 sends a write command (including the associated disk address and word count such that a cylinder boundary is not crossed) to the bus adaptor (66). The bus adaptor broadcasts the write command to all controllers (68) by passing appropriate CDBs via the SCSI buses to all of the controllers. The controllers will then all initiate a series of requests (70) over the SCSI buses on a byte-by-byte basis to the bus adaptor for the data that is to be stored. The data requests from different controllers typically will not appear synchronously. The bus adaptor includes logic circuitry 48 that is connected to all of the SCSI buses and will detect the requests as they appear. Logic circuitry 48 also has control outputs connected to control inputs of the respective shift registers 34, 36 to control the loading and unloading of the shift registers via the SCSI buses.

When all of the controllers have issued their requests, the bus adaptor will satisfy all requests simultaneously by sending unload signals from logic circuitry 48 to all shift registers 34, 36 to cause them to unload their bytes over all of the SCSI buses at the same time (72). Each

controller will continue to request data byte-by-byte until its buffer is full (or until the number of bytes in its buffer equals the word count). Each controller will then begin to write the data from its buffer onto the disk (74).

The different controllers will execute the write operations at asynchronous times which depend on the phase differences between the shaft locations of their respective disk drives. In the case when the word count exceeds the buffer capacity, as soon as the number of bytes remaining in its buffer falls below 8 kilobytes, each controller will request more data from the bus adaptor. These controller requests for more data typically will arrive asynchronously at the bus adaptor. The bus adaptor through its logic circuitry 48 will wait until all requests have been received before honoring them by delivering more bytes. In effect, the bus adaptor waits until the buffer in the controller which is the last to request more data, is drained to the 8 kilobyte level. Meanwhile, the buffer in the controller which was the first to request more data has been emptied below the 8 kilobyte level. Thereafter the bus adaptor will be delivering data bytes as soon as the first-to-request controller asks for them, thus ensuring that the buffer in the first-to-request controller is never fully emptied. The process continues until the number of data bytes delivered to each controller reaches the word count.

Referring to Fig. 4B, when data is to be retrieved, computer 10 sends a seek command (80) followed by a read command (82) both of which are broadcast by the bus adaptor to the controllers (84, 86) in a similar manner as for the data storage operation. Each controller will execute the seek command (88) and will begin to execute the read command by causing the disk drive to read bytes from the disk and to deliver them serially over the ESDI bus to be loaded in the controller buffer. The different controllers will typically execute the reading of data asynchronously. When a sector's worth of read data has been entered into the buffer, the controller will ask to have the buffer drained (90), but will continue the data reading and buffer filling. When the bus adaptor logic circuitry detects that all controllers are waiting to have their buffers drained, it begins draining them (92). Thereafter the bus adaptor will keep draining bytes from the controller buffers at the rate requested by the controller which was the first to request draining, thus ensuring that the first buffer is never full. The read operation continues until the number of bytes read from each disk drive equals the word count.

Should any one of the disk storage units 22 in each high-speed disk subsystem fail, the EDC unit will be able to correct erroneous bits received from the failed unit. The

entire failed unit can then be replaced by a working unit without interrupting the operation of the disk subsystem.

We thus provide a very high throughput rate storage system capable of serving a very wide, high-speed parallel I/O bus. Standard commercially available disk drives and controllers can be used without having to synchronize the rotations of the different disk drive spindles. The system is tolerant to the failure of one entire disk drive unit. Standard SCSI protocol commands are used to synchronize the overall operation of the system.

Other embodiments are feasible.

For example, referring to Fig. 5, the bus adaptor 18 could use a single EDC unit 32. In that case, the 64 lines of the KBUS are split into two sets of 32 lines each, both sets of which are connected to a multiplexer 30. When data is to be stored, multiplexer 30, during a single transfer cycle of the KBUS, first connects one set of 32 lines of the KBUS to the bidirectional WD/CD input of unit 32 and then connects the other set of 32 KBUS lines to the WD/CD input. The 32 lines connected to the WC/CD input are also connected respectively to the inputs of 32 eight-bit shift registers 34. During data retrieval, unit 32 delivers the 32 corrected data bits at the WD/CD output to a bank of 32 flip-flops (LFF) 40. Next the shift registers 34, 36 unload the second bits of their bytes to EDC unit 32, which in turn delivers the 32 corrected bits to a

bank of flip-flops (HFF) 42. At the same time the bank of flip-flops 40 shifts its 32 bits to a third bank of 32 flip-flops (HFF) 44. Then a KBUS transaction will deliver all 64 bits to the computer. Inverter 46 ensures that, when the computer is driving the KBUS, multiplexer 30 is active but banks 42, 44 are inactive, and otherwise multiplexer 30 is inactive and banks 42, 44 are able to deliver bits to the KBUS.

The input to inverter 46 is provided from logic circuitry 48 based on monitoring of the data bits passing back and forth to and from EDC unit 32. Circuitry 48 determines when it is appropriate to trigger the flip flop banks based on the appearances of the successive sets of 32 bits.

The number of KBUS lines served by each disk storage unit could be more or less than two.

CLAIMS

1. Apparatus for storing and retrieving digital data words and including data storage equipment adapted for storing each said data word on a mechanically-driven medium, said apparatus being characterised in that it comprises error correction means adapted for correcting errors in retrieved data words, which error correction means comprise: error correction circuitry adapted to generate error correction bits for said data words, and correction bit storage equipment adapted for storing said correction bits independently from said data word on a mechanically driven medium, said error correction circuitry being arranged operatively to correct errors in retrieved said data words using said error correction bits.

2. Apparatus according to Claim 1, further characterised in that said data storage equipment comprises data storage units arranged for operatively respectively storing different portions of each said data word, and said correction bit storage equipment comprises at least one correction bit storage unit.

3. Apparatus according to Claim 2, further characterised in that said error correction circuitry is arranged operatively to correct said data words in a manner that permits the failure of any single said storage unit.

4. Apparatus according to Claim 2, further characterised in that there are a plurality of correction bit storage units, and various said error correction bits are arranged operatively to be stored in different said correction bit storage units.

5. Apparatus according to Claim 2, further characterised in that no two said error correction bits for a given said data word are stored in a single said correction bit storage unit.

6. Apparatus according to Claim 2, further characterised in that said error correction circuitry comprises: an error correction unit arranged operatively to generate a set of said error correction bits for fewer than all of the bits in each said data word at a time, and routing circuitry adapted for routing the bits of said data word in successive groups to said error correction unit to generate a plurality of sets of said error correction bits.

7. Apparatus according to Claim 2, further characterised in that said error correction circuitry comprises: a plurality of error correction units each of which is arranged operatively to generate a set of error correction bits for a group of fewer than all of the bits in each said data word, and routing circuitry adapted for routing groups of the bits in each said data word in parallel to said error correction units to generate a plurality of sets of said data bits.

8. Apparatus according to Claims 6 or 7, wherein each said data word has 64 bits, and being further characterised in that each said error correction unit is arranged operatively to generate seven error correction bits for each group of 32 bits of each said data word.

9. Apparatus according to any of Claims 2 to 8, wherein said data words are delivered in succession via a parallel data bus for storage and retrieval, said system being further characterised in comprising an adaptor connected between said bus and said storage units and adapted for routing data words between said bus and said storage units.

10. Apparatus for storing digital data words and adapted for delivering them in succession to a parallel data bus when requested by a data processor connected to said bus, said apparatus being characterised in comprising: storage units each having a mechanically-driven medium, and a head for reading said medium, the storage units being so arranged that different portions of each said data word are stored in different said storage units, one said

storage unit having associated therewith a first delay time between when a particular said data word is requested and when the portion of said particular data word stored in said storage unit is ready to be delivered to said bus, another said storage unit having associated therewith a second delay time, different from said first delay time, between when said particular data word is requested and when the portion of said data word stored in said other storage unit is ready to be delivered to said bus; and an adaptor arranged for operative connection between said bus and said storage units and adapted to respond to said request from said data processor by causing said storage units to read said portions of each successive said word from said storage units.

11. Apparatus according to Claim 10, further characterised in that said adaptor is further adapted in operation to delay the delivery of any of said portions to said bus until all said portions are ready to be delivered.

12. Apparatus according to Claim 9 or Claim 10, further characterised in that an individual bus is provided between each said storage unit and said adaptor; in that each said storage unit is operatively arranged to send a drain command over its corresponding individual bus to said adaptor when said portion of said data stored in said storage unit is ready to be delivered; and in that said adaptor is operatively arranged to monitor the appearances of said drain commands on all of said individual buses, and, when said drain commands from all said storage units have appeared, to drain all said portions of said data word from said storage units and to deliver them simultaneously to said parallel data bus.

13. Apparatus according to Claim 10, further characterised in that said adaptor is arranged operatively to generate parallel error correction bits for each said data word, the apparatus being further arranged such that at least one of said error correction bits is stored on a storage unit that does not hold any said portion of the data word related to said error correction bits.

14. Apparatus according to Claim 13, further characterised in that said adaptor is further arranged operatively to cause said error correction bits to be read at a time corresponding to the time that the related said data word portions are read, and in that said adaptor comprises error correcting circuitry adapted for correcting bit errors in said data word based on said error correction bits.

15. Apparatus according to Claims 13 or 14, further characterised in being arranged such that different groups of said error correction bits are stored in different said storage units.

16. Apparatus according to Claim 15, further characterised in being arranged such that each said error correction bit for each said data word is stored in a separate said storage unit.

17. Apparatus according to any of Claims 10 to 16, further characterised in that it is arranged to store said data words when commanded by said data processor, said storage units each further comprising a head for writing on said medium, different portions of each said data word being arranged operatively to be stored in different said storage units, different said storage units having associated therewith different delay times between when a particular said word is delivered for storage and when said storage units are ready to store their respective said portions, and said adaptor being further arranged operatively to respond to said command from said data processor by delaying the delivery of any of said portions to said storage units until all said storage units are ready to store.

18. Apparatus according to Claim 17, further characterised in that each said storage unit comprises a controller having a buffer operatively arranged temporarily to hold bits that are to be written on said medium, and in that said adaptor is arranged operatively to effect the delivery of successive said data word portions to said storage units at a rate sufficient to ensure that none of the buffers in any said controller is ever empty.

19. Apparatus according to any of Claims 9 to 17, further characterised in that each said storage unit comprises a controller having a buffer operatively arranged temporarily to hold bits that have been read from said medium and are ready to be delivered, and in that said adaptor is arranged operatively to effect the delivery of successive said data word portions from said storage units at a rate sufficient to ensure that none of the buffers in any said controllers is ever full.

20. Apparatus according to any of Claims 9 to 19, further characterised in being arranged such that the portions of each said data word stored in different said storage units are equal.

21. Apparatus according to Claim 20, further characterised in that said portions are each two bits of said data word.

22. Apparatus according to any of Claims 10 to 21, further characterised in that each said individual bus conforms to the SCSI protocol.

23. Apparatus according to any of Claims 1 to 7 or 9 to 22, further characterised in that said apparatus is adapted for storing and retrieving a data word having more than 32 bits therein, and preferably 256 bits therein.

24. Apparatus according to Claims 8 or 10 or any claim appendent thereto, further characterised in being arranged such that said data words are delivered to said parallel data bus at a rate of at least 4 megabits per second per line of said bus.

25. Apparatus according to Claims 2 or 10, or any claim appendent thereto, further characterised in that each said storage unit comprises a magnetic disk drive.

0201330

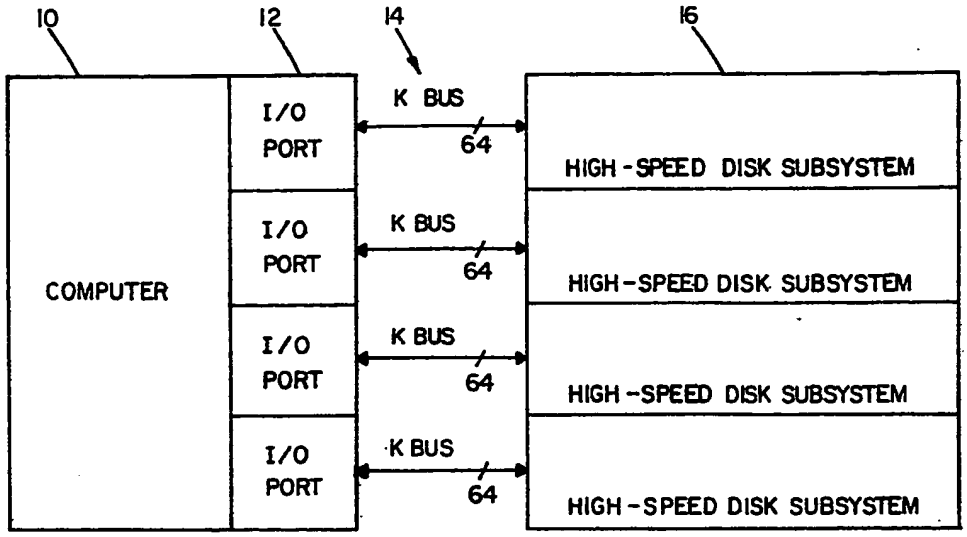


FIG 1

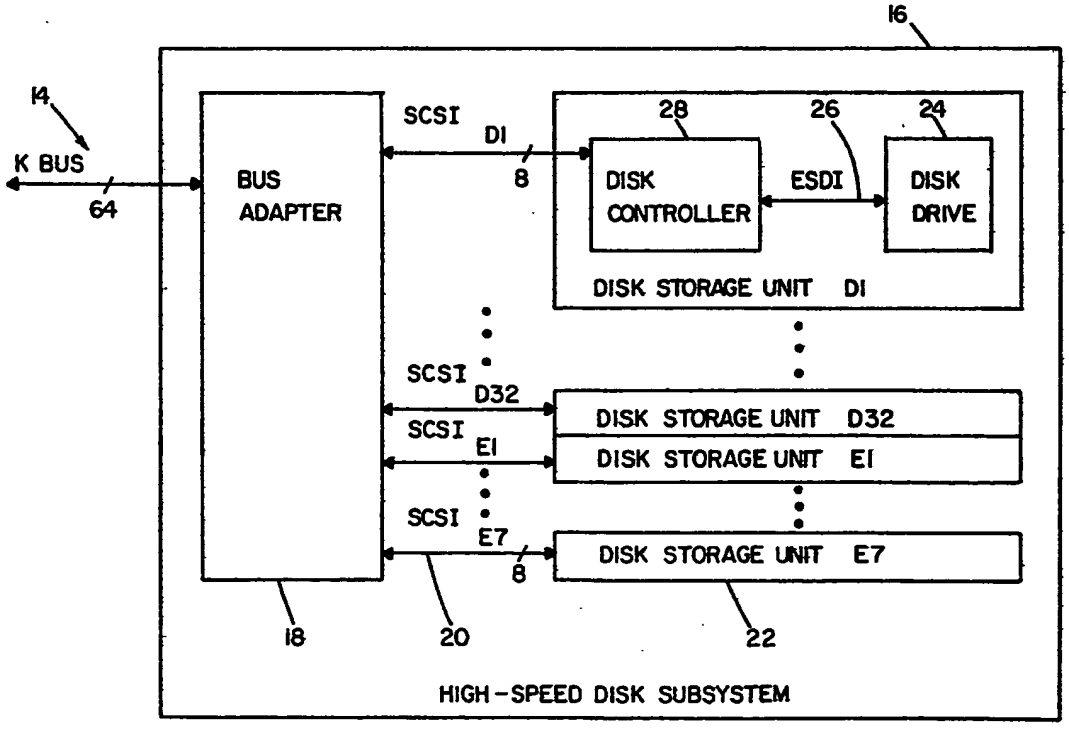


FIG 2

0201330

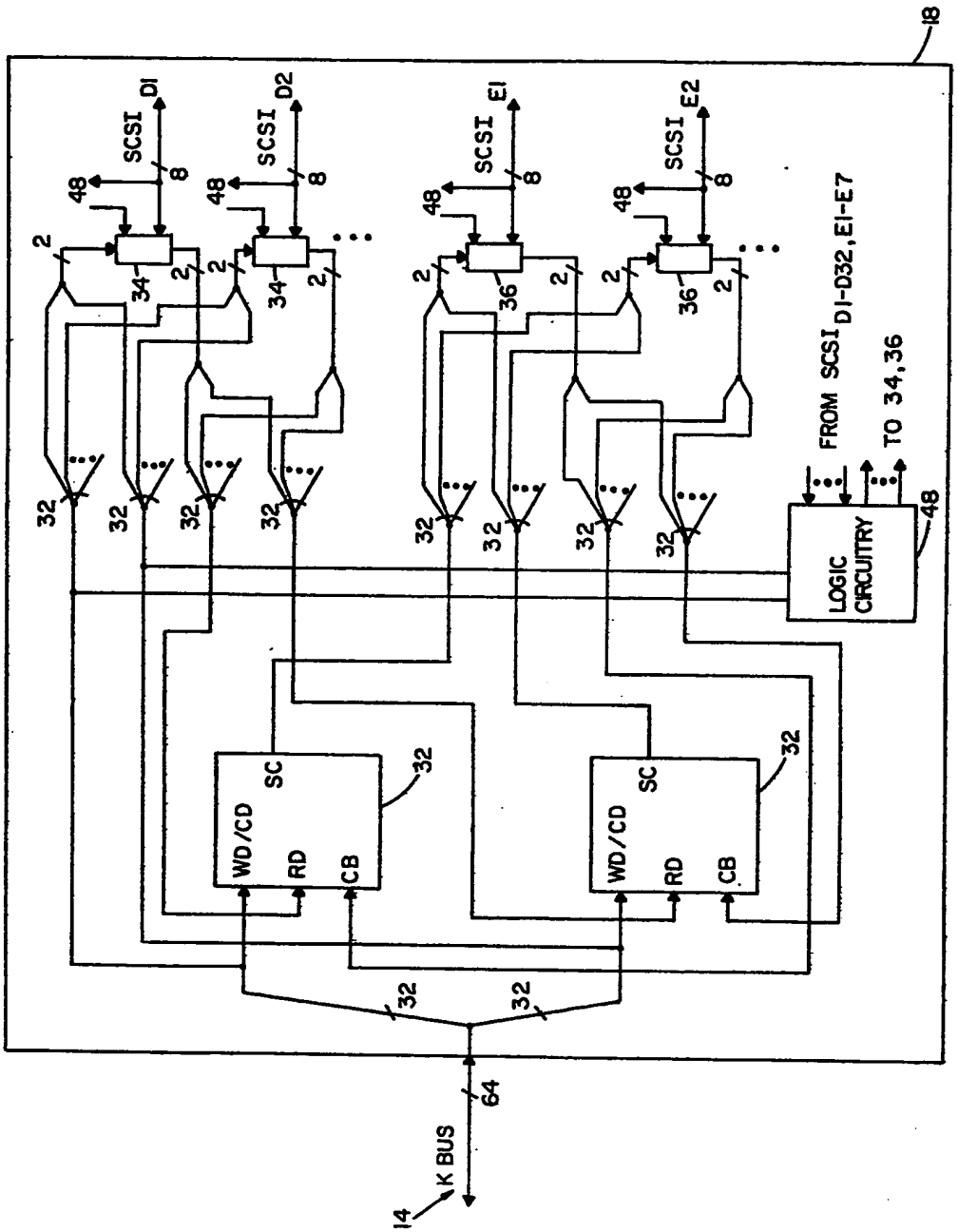


FIG 3

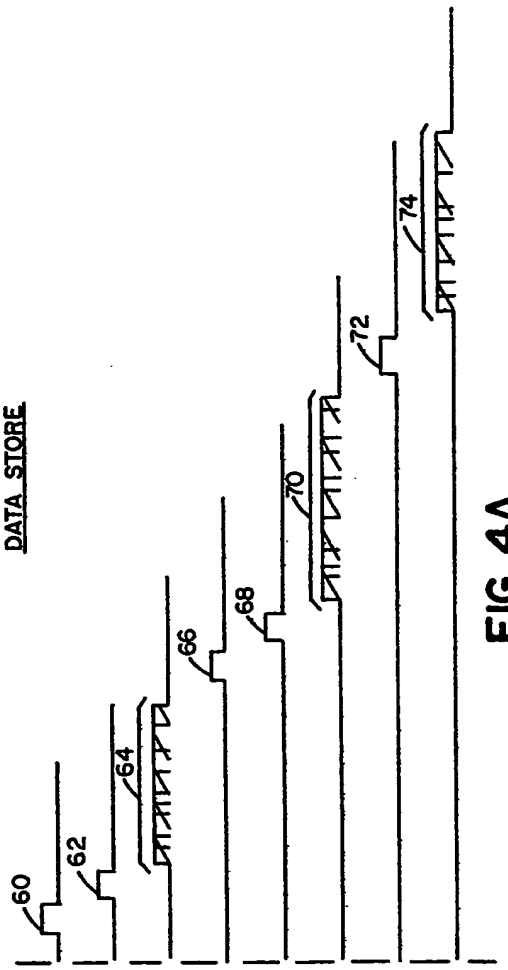

DATA STORE
60
62
64
66
68
70
72
74
K BUS SEEK COMMAND
SCSI BUSES SEEK COMMANDS
SEEK COMMANDS EXECUTED BY ALL CONTROLLERS
K BUS WRITE COMMAND
SCSI BUSES WRITE COMMANDS
SCSI BUSES DATA REQUESTS
BYTES SENT OVER SCSI BUSES
CONTROLLERS EXECUTE WRITES

FIG 4A

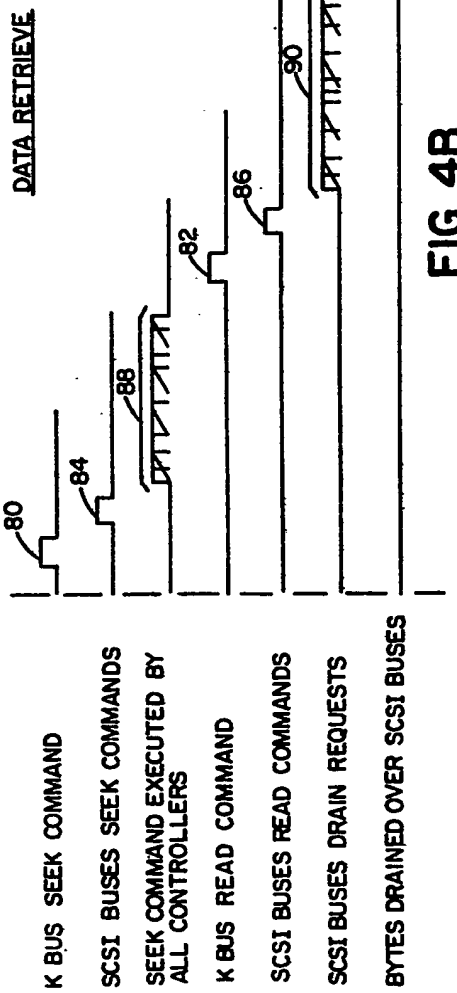

DATA RETRIEVE
80
84
88
82
86
90
92
K BUS SEEK COMMAND
SCSI BUSES SEEK COMMANDS
SEEK COMMAND EXECUTED BY ALL CONTROLLERS
K BUS READ COMMAND
SCSI BUSES READ COMMANDS
SCSI BUSES DRAIN REQUESTS
BYTES DRAINED OVER SCSI BUSES

FIG 4B

0201330

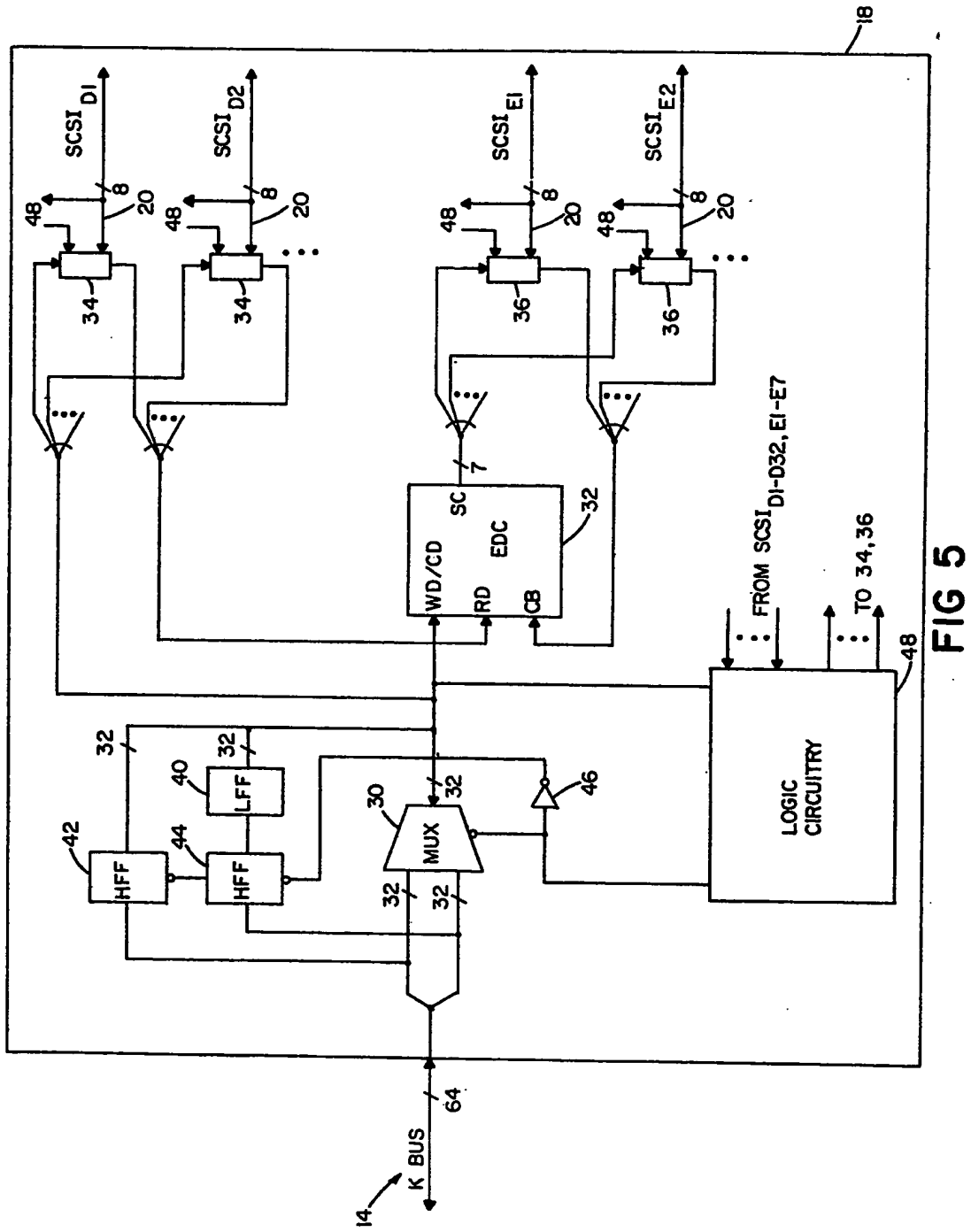


FIG 5